# Benchmarking and scaling studies of a pseudospectral code Tarang for turbulence simulations

Mahendra K. Verma[a,*], Anando Chaterjee[a], Supriyo Paul[b], Sandeep Reddy[c], Rakesh K. Yadav[a], Mani Chandra[a], Ravi Samtaney[d]

[a]*Department of Physics, Indian Institute of Technology Kanpur, India 208016*
[b]*Computational Fluid Dynamics Team, Centre for Development of Advanced Computing, Pune, India 411007*
[c]*Department of Mechanical Engineering, Indian Institute of Technology Kanpur, India 208016*
[d]*Department of Mechanical Engineering, King Abdullah University of Science and Technology Thuwal, Kingdom of Saudi Arabia 23955-6900*

**Abstract**

Tarang is a general-purpose pseudospectral parallel code for simulating flows involving fluids, magnetohydrodynamics, and Rayleigh-Bénard convection in turbulence and instability regimes. In this paper we present code validation and benchmarking results of Tarang. We performed our simulations on $1024^3$, $2048^3$, and $4096^3$ grids using the *HPC system* of IIT Kanpur and *Shaheen* of KAUST. We observe good "weak" and "strong" scaling for Tarang on these systems.

*Keywords:* Pseudospectral Method; Direct Numerical Simulations; High-performance Computing

## 1. Introduction

A typical fluid flow is random or chaotic in the turbulent and instability regimes. Therefore we need to employ accurate numerical schemes for simulating such flows. A pseudospectral algorithm [1, 2] is one of the most accurate methods for solving fluid flows, and it is employed for performing direct numerical simulations of turbulent flows, as well as for critical applications like weather predictions and climate modelling. Yokokawa et al. [3, 4], Donzis et al. [5], and Pouquet et al. [6] have performed spectral simulations on some of the largest grids (e.g., $4096^3$).

We have developed a general-purpose flow solver named Tarang (synonym for waves in Sanskrit) for turbulence and instability studies. Tarang is a parallel and modular code written in object-oriented language C++. Using Tarang, we can

---

*Corresponding author
*Email address:* mkv@iitk.ac.in (Mahendra K. Verma)

solve incompressible flows involving pure fluid, Rayleigh-Bénard convection, passive and active scalars, magnetohydrodynamics, liquid-metals, etc. Tarang is an open-source code and it can be downloaded from `http://turbulence.phy.iitk.ac.in`. In this paper we will describe some details of the code, scaling results, and code validation performed on Tarang.

## 2. Salient features of Tarang

The basic steps of Tarang follow the standard procedure of pseudospectral method [1, 2]. The Navier-Stokes and related equations are numerically solved given an initial condition of the fields. The fields are time-stepped using one of the time integrators. The nonlinear terms, e.g. $\mathbf{u} \cdot \nabla \mathbf{u}$, transform to convolutions in the spectral space, which are very expensive to compute. Orszag devised a clever scheme to compute the convolution in an efficient manner using Fast Fourier Transforms (FFT) [1, 2]. In this scheme, the fields are transformed from the Fourier space to the real space, multiplied with each other, and then transformed back to the Fourier space. Note that the spectral transforms could involve Fourier functions, sines and cosines, Chebyshev polynomials, spherical harmonics, or a combination of these functions depending on the boundary conditions. For details the reader is referred to standard references, e.g., the books by Boyd [1] and Canuto et al. [2]. Some of the specific choices made in Tarang are as follow:

(a) In the turbulent regime, the two relevant time scales, the large-eddy turnover time and the small-scale viscous time, are very different (order of magnetic apart). To handle this feature, we use the "exponential trick" that absorbs the viscous term using a change of variable [2].
(b) We use the fourth-order Runge-Kutta scheme for time stepping. The code however has an option to use the Euler and the second-order Runge-Kutta schemes as well.
(c) The code provides an option for dealiasing the fields. The 3/2 rule is used for dealiasing [2].
(d) The wavenumber components $k_i$ are

$$k_i = \frac{2\pi}{L_i} n_i \tag{1}$$

where $L_i$ is the box dimension in the $i$-th direction, and $n_i$ is an integer. We use parameters

$$\text{kfactor}_i = \frac{2\pi}{L_i} \tag{2}$$

to control the box size, especially for Rayleigh-Bénard convection. Note that typical spectral codes take $\text{kfactor}_i = 1$, or $k_i = n_i$.

The parallel implementation of Tarang involved parallelization of the spectral transforms and the input-output operations, as described below.

2

### 3. Parallelization Strategy

A pseudospectral code involves forward and inverse transforms between the spectral and real space. In a typical pseudospectral code, these operations take approximately 80% of the total time. Therefore, we use one of the most efficient parallel FFT routines, FFTW (Fastest Fourier Transform in the West) [7], in Tarang. We adopt FFTW's strategy for dividing the arrays. If $p$ is the number of available processors, we divide each of the arrays into $p$ "slabs". For example, a complex array $A(N_1, N_2, N_3/2 + 1)$ is split into $A(N_1/p, N_2, N_3/2 + 1)$ segments, each of which is handled by a single processor. This division is called "slab decomposition". The other time-consuming tasks in Tarang are the input and output (I/O) operations of large data sets, and the element-by-element multiplication of arrays. The data sets in Tarang are massive, for example, the data size of a $4096^3$ fluid simulation is of the order of 1.5 terabytes. For I/O operations, we use an efficient and parallel library named HDF5 (Hierarchical Data Format-5). The third operation, element-by-element multiplication of arrays, is handled by individual processors in a straightforward manner.

Tarang has been organized in a modular fashion, so the spectral transforms and I/O operations were easily parallelized. For a periodic-box, we use the parallel FFTW library itself. However, for the mixed transforms (e.g., sine transform along $x$, and Fourier transform along $yz$ plane), we parallelize the transforms ourselves using one- and two-dimensional FFTW transforms.

An important aspect of any parallel simulation code is its scalability. We tested the scaling of FFTW and Tarang by performing simulations on $1024^3$, $2048^3$, and $4096^3$ grids with variable number of processors. The simulations were performed on the *HPC system* of IIT Kanpur and *Shaheen* supercomputer of King Abdullah University of Science and Technology (KAUST). The *HPC system* has 368 compute nodes connected via a 40 Gbps Qlogic Infiniband switch with each node containing dual Intel Xeon Quadcore C5570 processor and 48GB of RAM. Its peak performance (Rpeak) is approximately 34 teraflops (tera floating point operations per second). *Shaheen* on the other hand is a 16-rack IBM BlueGene/P system with 65536 cores and 65536 GB of RAM. *Shaheen*'s peak performance is approximately 222 teraflops.

For parallel FFT with slab decomposition, we compute the time taken per step (forward+backward transform) on *Shaheen* for several large $N^3$ grids. The results displayed in Fig. 1 demonstrate an approximate linear scaling (called "strong scaling"). Using the fact that each forward plus inverse FFT involves $5N^3 \log N^3$ operations for single precision computations [7], the average FFT performance per core on *Shaheen* is approximately 0.3 gigaflops, which is only 8% of its peak performance. Similar efficiency is observed for the *HPC system* as well, whose cores have rating of approximately 12 gigaflops. The aforementioned loss of efficiency is consistent with the other FFT libraries, e.g, p3dfft [8]. Also note that an increase in the data size and number of processors (resources) by a same amount takes approximately the same time (see Fig. 1). For example, FFT of a $1024^3$ array using 128 processors, as well as that of a $2048^3$ array on 1024 processors, takes approximately 4 seconds. Thus our implementation of
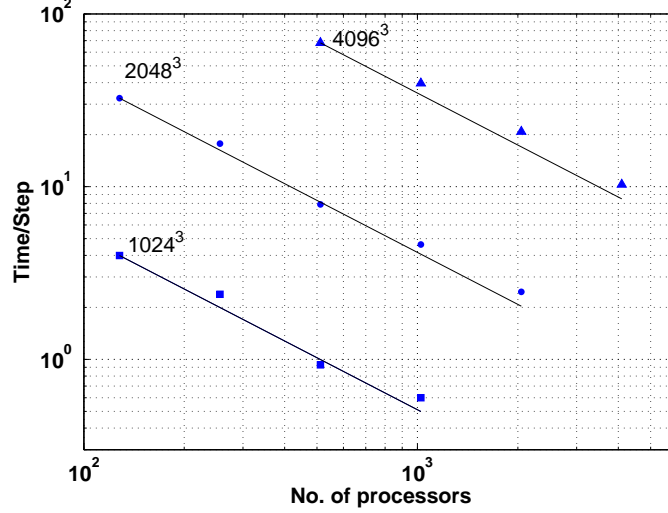
Figure 1: Scaling of parallel FFT on *Shaheen* for $1024^3, 2048^3$ and $4096^3$ grids with single precision computation. The straight lines represent the ideal linear scaling.

FFT shows good "weak scaling" as well.

We also test the scaling of Tarang on *Shaheen* and the *HPC system*. Figs. 2 and 3 exhibit the scaling results of fluid simulations performed on these systems. Fig. 4 shows the scaling results for magnetohydrodynamics (MHD) simulation on Shaheen. These plots demonstrate *strong scaling* of Tarang, consistent with the aforementioned FFT scaling. Sometimes we observe a small loss of efficiency when $N = p$. We also observe approximate *weak scaling* for Tarang on both *Shaheen* and the *HPC system*.

A critical limitation of the "slab decomposition" is that the number of processor cannot be more than $N_1$. This limitation can be overcome in a new scheme called "pencil decomposition" in which the array $A(N_1, N_2, N_3/2 + 1)$ is split into $A(N_1/p_1, N_2/p_2, N_3/2 + 1)$ pencils where the total number of processors $p = p_1 \times p_2$ [8]. We are in the process of implementing "pencil decomposition" on Tarang. In this paper we will focus only on the "slab decomposition".

After the above discussion on parallelization of the code, we will discuss code validation, and time and space complexities for simulations of fluid turbulence, Rayleigh-Bénard convection, and magnetohydrodynamic turbulence.

## 4. Fluid turbulence

The governing equations for incompressible fluid turbulence are

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}^u, \tag{3}$$
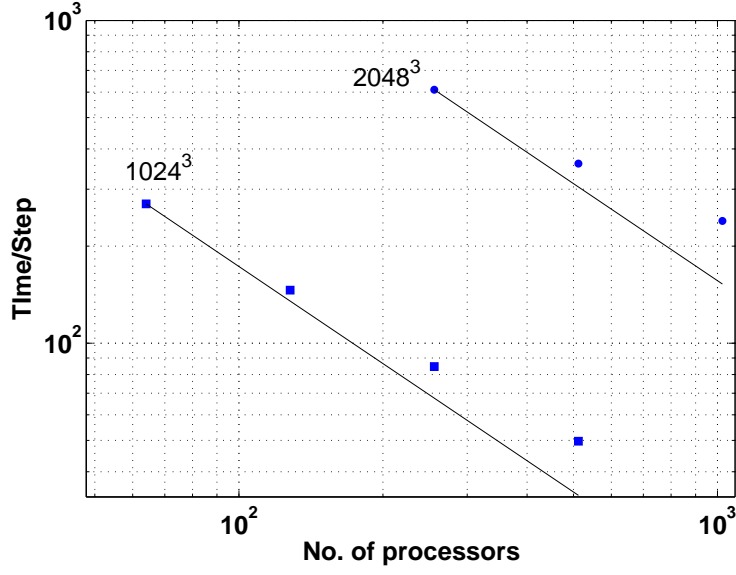$$\nabla \cdot \mathbf{u} = 0, \tag{4}$$

4

Figure 2: Scaling of Tarang's fluid solver on *Shaheen* for $1024^3$ and $2048^3$ grids with single precision computation. The straight lines represent the ideal linear scaling.
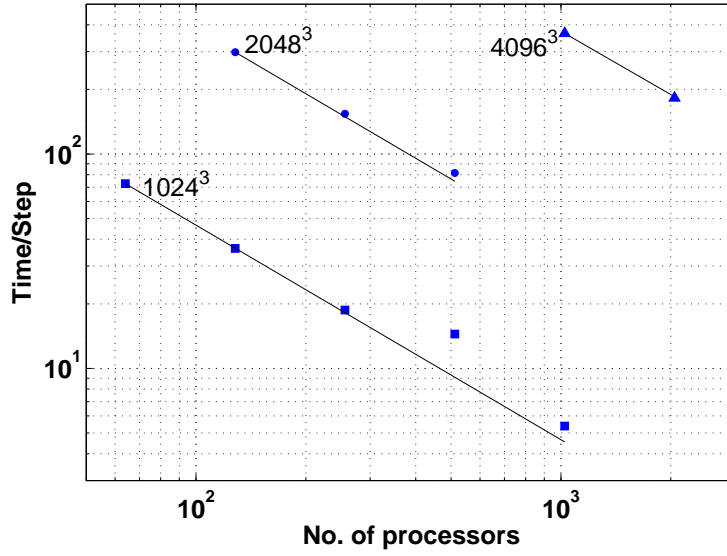


Figure 3: Scaling of Tarang's fluid solver on the *HPC system* of IIT Kanpur for $1024^3$, $2048^3$, and $4096^3$ grids with single precision computation. The straight lines represent the ideal linear scaling.
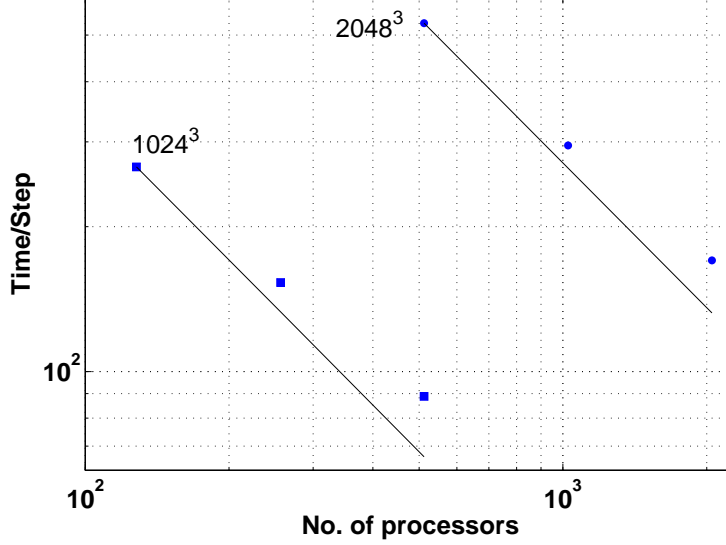
Figure 4: Scaling of Tarang's magnetohydrodynamic (MHD) solver on *Shaheen* for $1024^3$ and $2048^3$ grids with single precision computation. The straight lines represent the ideal linear scaling.

where **u** is the velocity field, $p$ is the pressure field, $\nu$ is the kinematic viscosity, and $\mathbf{F}^u$ is the external forcing. For studies on homogeneous and isotropic turbulence, simulations are performed on high-resolution grids (e.g, $2048^3, 4096^3$) with a periodic boundary condition. The resolution requirement is stringent due to $N \sim Re^{3/4}$ relation; for $Re = 10^5$, the required grid resolution is approximately $5600^3$, which is quite challenging even for modern supercomputers.

Regarding the space complexity of a forced fluid turbulence simulation, Tarang requires 15 arrays (for $\mathbf{u}(\mathbf{k}), \mathbf{u}(\mathbf{r}), \mathbf{F}^u(\mathbf{k}), \mathbf{nlin}(\mathbf{k})$, and three temporary arrays), which translates to approximately 120 gigabytes (8 terabytes) of memory for $1024^3$ ($4096^3$) double-precision computations. Here **k** and **r** represent the wavenumbers and the real space coordinates respectively. The requirement is halved for a simulation with single precision. Regarding the time requirement, each numerical step of the fourth-order Runge-Kutta (RK4) scheme requires $9 \times 4$ FFT operations. The factor 9 is due to the 3 inverse and 6 forward transforms performed for each of the four RK4 iterates. Therefore, for every time step, all the FFT operations require $36 \times 2.5 \times N^3 \log_2(N^3)$ multiplications for a single precision simulation [7], which translates to approximately 2.9 (185) tera floating-point operations for $1024^3$ ($4096^3$) grids. The number of operations for double-precision computation is twice of the above estimate. On 128 processors on *HPC system*, a fluid simulation with single-precision takes approximately 36 seconds (see Fig. 3), which corresponds to per core performance of approx-

imately 0.68 gigaflops. This is only 6% of the peak performance of the cores, which is consistent with the efficiency of FFT operations discussed in Section 3. Also note that the solver also involves other operations, e.g., element-by-element array multiplication, but these operations take only a small fraction of the total time.

We can also estimate the total time required to perform a $4096^3$ fluid simulation. A typical fluid turbulence would require 5 eddy turnover time with $dt \approx 5 \times 10^{-4}$, which corresponds to $10^4$ time steps for the simulations. So the total floating point operations required for this single-precision simulation is $185 \times 10^4$ tera floating-point operations for the FFT itself. Assuming 5% efficiency for FFT, and FFTs share being 80% of the total time, the aforementioned fluid simulation will take approximately 128 hours on a 100 teraflop cluster.

We perform code validation of the fluid solver using Kolmgorov's theory [9] for the third-order structure function, according to which

$$S_3^{||}(r) = \langle \{u_{||}(\mathbf{x} + \mathbf{r}) - u_{||}(\mathbf{x})\}^3 \rangle = -\frac{4}{5}\epsilon r \tag{5}$$

where $\epsilon$ is the energy flux in the inertial range, and $\langle ... \rangle$ represents ensemble averaging (here spatial averaging). We compute the structure function $S_3^{||}(r)$, as well as $S_5^{||}(r)$, $S_7^{||}(r)$, and $S_9^{||}(r)$ for the steady-state dataset of a fluid simulation on a $1024^3$ grid. The computed values of $S_q^{||}(r)$ are illustrated in Fig. 5 that shows a good agreement with Kolmogorov's theory.



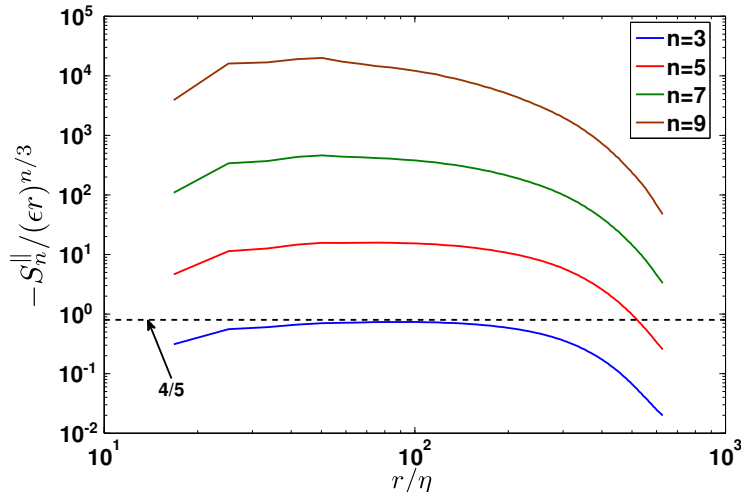Figure 5: Plots of the normalized odd-order structure functions $-S_n^{||}(r)/(\epsilon r)^{n/3}$ vs. $r/\eta$ for a fluid simulation using Tarang. Here $\epsilon$ is the energy flux, and $\eta$ is the Kolmogorov scale.

After the discussion on fluid solver, we move on to the module for solving Rayleigh-Bénard convection.

## 5. Rayleigh-Bénard convection

Rayleigh-Bénard convection (RBC) is an idealized model of convection in which fluid is subjected between two plates that are separated by a distance $d$, and are maintained at temperatures $T_0$ and $T_0 - \Delta$. The equations for the above fluid under Boussinesq approximations are

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla \sigma}{\rho_0} + \alpha g \theta \hat{z} + \nu \nabla^2 \mathbf{u}, \tag{6}$$

$$\partial_t \theta + (\mathbf{u} \cdot \nabla)\theta = \frac{\Delta}{d} u_z + \kappa \nabla^2 \theta, \tag{7}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{8}$$

where $\theta$ and $\sigma$ are the temperature and pressure fluctuations from the steady conduction state ($T = T_c + \theta$ with $T_c$ as the conduction temperature profile), $\hat{z}$ is the buoyancy direction, $\Delta$ is the temperature difference between the two plates, $\nu$ is the kinematic viscosity, and $\kappa$ is the thermal diffusivity. We solve the nondimensionalized equations, which are obtained using $d$ as the length scale, $\kappa/d$ as the velocity scale, and $\Delta$ as the temperature scale:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla \sigma + RP\theta \hat{z} + P\nabla^2 \mathbf{u}, \tag{9}$$

$$\frac{\partial \theta}{\partial t} + (\mathbf{u} \cdot \nabla)\theta = u_3 + \nabla^2 \theta. \tag{10}$$

Here the two important nondimensional parameters are the Rayleigh number $R = \alpha g \Delta d^3 / \nu \kappa$, and the Prandtl number $P = \nu/\kappa$. In Tarang we can apply the free-slip boundary condition for the velocity fields at the horizontal plates, i.e.,

$$u_3 = \partial_z u_1 = \partial_z u_2 = 0 \qquad \text{for } z = 0, 1, \tag{11}$$

and isothermal boundary condition on the horizontal plates

$$\theta = 0 \qquad \text{for } z = 0, 1, \tag{12}$$

Periodic boundary conditions are applied to the vertical boundaries.

The number of arrays required for a RBC simulation is 18 (15 for fluids plus three for $\theta(\mathbf{k}), \theta(\mathbf{r}), \mathbf{nlin}^{\theta}$). Thus the memory requirement for RBC is (18/15) times that for the fluid simulation. Regarding the time complexity, the number of FFT operations required per time step is $13 \times 4$ FFT operations (4 inverse + 9 forward transforms per RK4 step). As a result, the total time requirement for a RBC simulation is (13/9) times the respective fluid simulation.

For code validation of Tarang's RBC solver, we compare the Nusselt number $Nu = 1 + \langle u_3 \theta \rangle$ computed using Tarang with that computed by Thual [10] for two-dimensional free-slip box. The analysis is performed for the steady-state dataset. The comparative results shown in Table 1 illustrate excellent agreement between the two runs. We also compute the Nusselt number for a three-dimensional flow with $Pr = 6.8$ and observe that $Nu = (0.27 \pm 0.04)(PrRa)^{0.27 \pm 0.01}$ [11], which is in good agreement with earlier experimental and numerical results.

Table 1: Verification of Tarang against Thual's [10] 2D RBC simulations. We compare Nusselt numbers ($Nu$) computed in our simulations on a $64^2$ grid against Thual's simulations on $16^2$ (THU1), $32^2$ (THU2), and $64^2$ (THU3) grids. All $Nu$ values tabulated here are for the Prandtl number of 6.8.

| $r$ | THU1 | THU2 | THU3 | Tarang |
|-----|-------|-------|-------|--------|
| 2   | 2.142 | –     | –     | 2.142  |
| 3   | 2.678 | –     | –     | 2.678  |
| 4   | 3.040 | 3.040 | –     | 3.040  |
| 6   | 3.553 | 3.553 | –     | 3.553  |
| 10  | 4.247 | 4.244 | –     | 4.243  |
| 20  | 5.363 | 5.333 | 5.333 | 5.333  |
| 30  | 6.173 | 6.105 | 6.105 | 6.105  |
| 40  | 6.848 | 6.742 | 6.740 | 6.740  |
| 50  | 7.441 | 7.298 | 7.295 | 7.295  |

Using the RBC module of Tarang, we also studied the energy spectra and fluxes of the velocity and temperature fields [12], the Nusselt number scaling [11], and chaos and bifurcations near the onset of convection [13, 14].

In the next section we will discuss the results of the MHD module of Tarang.

## 6. Magnetohydrodynamic turbulence and dynamo

The equations for the incompressible MHD turbulence [15] are

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + (\mathbf{B} \cdot \nabla)\mathbf{B} + \nu\nabla^2\mathbf{u} + \mathbf{F}^u, \qquad (13)$$
$$\partial_t \mathbf{B} + (\mathbf{u} \cdot \nabla)\mathbf{B} = (\mathbf{B} \cdot \nabla)\mathbf{u} + \eta\nabla^2\mathbf{B} + \mathbf{F}^B, \qquad (14)$$
$$\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{B} = 0, \qquad (15)$$

where $\mathbf{u}$, $\mathbf{B}$ and $p$ are the velocity-, magnetic-, and pressure (thermal+magnetic) fields respectively, $\nu$ is the kinematic viscosity, and $\eta$ is the magnetic diffusivity. The $\mathbf{F}^u$ and $\mathbf{F}^B$ are external forcing terms for the velocity and magnetic fields respectively. Typically, $\mathbf{F}^B = 0$, but Tarang implements $\mathbf{F}^B$ for generality. The magnetic field $\mathbf{B}$ can be separated into its mean $\mathbf{B}_0$ and fluctuations $\mathbf{b}$: $\mathbf{B} = \mathbf{B}_0 + \mathbf{b}$. The number of nonlinear terms in the above equations is four whose computation requires 27 FFTs. However, the number of FFT computations in terms of the Elsasser variables $\mathbf{z}^\pm = \mathbf{u} \pm \mathbf{b}$ is only 15, thus saving significant computing time. We use

$$(\mathbf{u} \cdot \nabla)\mathbf{u} - (\mathbf{B} \cdot \nabla)\mathbf{B} = (\mathbf{z}^- \cdot \nabla)\mathbf{z}^+ + (\mathbf{z}^+ \cdot \nabla)\mathbf{z}^-, \qquad (16)$$
$$(\mathbf{u} \cdot \nabla)\mathbf{B} - (\mathbf{B} \cdot \nabla)\mathbf{u} = (\mathbf{z}^- \cdot \nabla)\mathbf{z}^+ - (\mathbf{z}^+ \cdot \nabla)\mathbf{z}^- \qquad (17)$$

to compute the nonlinear terms. Thus, the time requirement for a MHD simulation would be around 15/9 times that for the fluid simulation. In Fig. 4 we
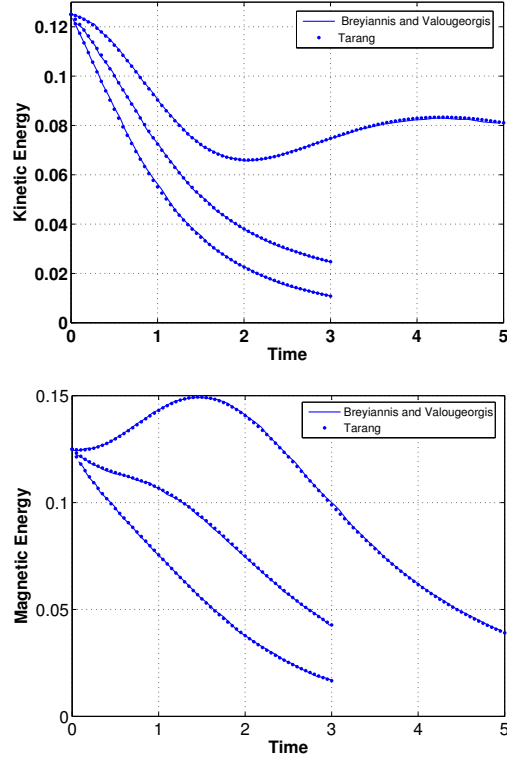
9

Figure 6: Time evolution of total kinetic energy (top panel) and total magnetic energy (bottom panel) for a decaying MHD simulation with Taylor-Green vortex as an initial condition. Blue dots are Tarang's data points, while the solid lines are the lattice simulation result of Breyiannis and Valougeorgis [16]. The three different curves reported here are for $\nu = \eta = 0.01$, 0.05, 0.1 from top to bottom.

plot the time taken per step for different set of processors on *Shaheen*. The results are consistent with the above estimates. Regarding the space complexity, an MHD simulation requires 27 arrays for storing $\mathbf{u}(\mathbf{k}), \mathbf{B}(\mathbf{k}), \mathbf{B}(\mathbf{r}), \mathbf{u}(\mathbf{r})$, $\mathbf{F}^u(\mathbf{k}), \mathbf{F}^B(\mathbf{k}), \mathbf{nlin}^u(\mathbf{k}), \mathbf{nlin}^B(\mathbf{k})$, and three temporary fields. Hence the memory requirement for a MHD simulation is 27/15 times that of a fluid simulation.

We perform code validation of Tarang's MHD module using the results of Breyiannis and Valougeorgis's [16] lattice kinetic simulations of three-dimensional decaying MHD. Following Breyiannis and Valougeorgis, we solve the MHD equations inside a cube with periodic boundary conditions on all directions, and with a Taylor-Green vortex (given below) as an initial condition,

$$\mathbf{u} \quad = \quad [\sin(x)\cos(y)\cos(z), -\cos(x)\sin(y)\cos(z), 0]\,, \qquad (18)$$

$$\mathbf{B} \quad = \quad [\sin(x)\sin(y)\cos(z), \cos(x)\cos(y)\cos(z), 0]\,. \qquad (19)$$

This Taylor-Green vortex is then allowed to evolve freely. The simulation box is discretized using $32^3$ grid points.

The results of this test case for different parameter values ($\nu = \eta = 0.01$, 0.05, 0.1) are presented in Fig. 6. The top and bottom panels exhibit the time evolution of the total kinetic- and magnetic energies respectively. Tarang's data points, illustrated using blue dots, are in excellent agreement with Breyiannis and Valougeorgis' results [16], which is represented using solid lines. We thus verify the MHD module of Tarang.

We have used Tarang to perform extensive simulations of dynamo transition under the Taylor-Green forcing [17, 18]. Using Tarang, we have also computed the magnetic and kinetic energy spectra, various energy fluxes [15], and shell-to-shell energy transfers for MHD turbulence; these results would be presented in a subsequent paper.

In addition to the fluid, MHD, and Rayleigh-Bénard convection solvers, Tarang has modules for simulating rotating turbulence, passive and active scalars, liquid metal flows, rotating convection [19], and Kolmogorov flow.

## 7. Conclusions

In this paper we describe salient features and code validation of Tarang. Tarang passes several validation tests performed for fluid, Rayleigh-Bénard convection, and magnetohydrodynamic solvers. We also report scaling analysis of Tarang and show that it exhibits excellent strong- and weak scaling up to several thousand processors. Tarang has been used for studying Rayleigh-Bénard convection, dynamo, and magnetohydrodynamic turbulence. It has been ported to various computing platforms including the *HPC system* of IIT Kanpur, *Shaheen* of KAUST, *Param Yuva* of the Centre for Advanced Computing (Pune), and *EKA* of the Computational Research Laboratory (Pune).

## References

[1] Boyd JP. Chebyshev and Fourier Spectral Methods. New York: Dover Publishers; 2001.

[2] Canuto C, Hussaini MY, Quarteroni A, Zhang TA. Spectral Methods in Fluid Turbulence. Berlin: Springer-Verlag; 1998.

[3] Yokokawa M, Itakura K, Uno A, Ishihara T, Kaneda Y. 16.4-TFlops direct numerical simulation of turbulence by a Fourier spectral method on the Earth Simulator. Tech. Rep.; dspace.itri.aist.go.jp; 2002.

[4] Kaneda Y, Ishihara T, Yokokawa M, Itakura K, Uno A. Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box. Phys Fluids 2003;15:L21.

[5] Donzis DA, Sreenivasan KR, Yeung PK. The Batchelor spectrum for mixing of passive scalars in isotropic turbulence. Flow Turbul Combust 2010;85:549.

[6] Pouquet A, Baerenzung J, Mininni PD, Rosenberg D, Thalabard S. Rotating helical turbulence: three-dimensionalization or self-similarity in the small scales?. Journal of Physics: Conference Series 2011;318:042015.

[7] Frigo M, Johnson SG. The design and implementation of FFTW3. Proceedings of the IEEE 2005;93(2):216–231;

[8] Parallel three-dimensional Fast Fourier Transforms (p3dfft) library. http://code.google.com/p/p3dfft; http://www.fftw.org/. 2008.

[9] Kolmogorov AN. Local structure of turbulence in incompressible viscous fluid for very large Reynolds number. Dokl Akad Nauk SSSR 1941;30:9–13.

[10] Thual O. Zero-Prandtl-number convection. J Fluid Mech 1992;240:229.

[11] Verma MK, Mishra PK, Pandey A, Paul S. Scalings of field correlations and heat transport in turbulent convection. Phys Rev E 2012;85:016310.

[12] Mishra PK, Verma MK. Energy spectra and fluxes for Rayleigh-Bénard convection. Phys Rev E 2010;81:056316.

[13] Pal P, Wahi P, Paul S, Verma MK, Kumar K, Mishra PK. Bifurcation and chaos in zero-Prandtl-number convection. EPL 2009;87:54003.

[14] Paul S, Pal P, Wahi P, Verma MK. Dynamics of zero-Prandtl number convection near the onset. Chaos 2011;21:023118.

[15] Verma MK. Statistical theory of magnetohydrodynamic turbulence: Recent results. Phys Rep 2004;401:229–380.

[16] Breyiannis G, Valougeorgis D. Lattice kinetic simulations of 3-d MHD turbulence. Computers & Fluids 2006;35(8–9):920 – 924.

[17] Yadav R, Chandra M, Verma MK, Paul S, Wahi P. Dynamo transition under Taylor-Green forcing. EPL 2010;91:69001.

[18] Yadav R, Verma MK, Wahi P. Bistability and chaos in the Taylor-Green dynamo. Phys Rev E 2012;85:036301.

[19] Pharasi HK, Kannan R, Kumar K, Bhattacharjee JK. Turbulence in rotating Rayleigh-Bénard convection in low-Prandtl-number fluids. Phys Rev E 2011;84:047301.